Student Exam No. ____________________

GANPAT UNIVERSITY

B. Tech. Semester: VI Computer Engineering/Information Technology

CBCS Regular Examination April - June 2016

2CE604/2IT604: Design & Analysis of Algorithms

Time: 3 Hours **Total Marks: 70**

Instructions: 1. Attempt all questions.
2. Figures to the right indicate full marks.
3. Each section should be written in a separate answer book.
4. Assume necessary data when needed.

SECTION – I

Que.1 [A] Calculate time complexity for following algorithm using tabular method. [3]

```
Algorithm complexity(){
  for(int i=1;i<=n;i++){
    for(int j=1;j<n;j=j*3){
      printf("Hello world");
    }
  }
}
```

[B] Express complexity of following functions using theta(Θ) notation. Clearly indicates value of constants c_1, c_2 and n_0. [9]

i. $f(n) = 3n^3 2^n + 5n^2 3^n$

ii. $f(n) = 2n^2 2^n + n \lg n$

iii. $f(n) = n^{2.5} + n \lg n$

OR

Que.1 [A] Briefly explain time and space complexity of an algorithm. Write recursive algorithm to find factorial of a given number and find its space complexity. [6]

[B] Check whether following statements are true or false. Justify your answer. [6]

i. $n^2 \lg n \neq \Theta(n^2)$

ii. $n^2 / \lg n \neq \Theta(n^2)$

Que.2 [A] Solve the following recurrence using homogeneous recurrence method. [6]

$$F_n = \begin{cases} n & ,\text{if } n = 0 \text{ or } 1 \\ F_{n-1} + F_{n-2} & ,\text{otherwise} \end{cases}$$

[B] Solve the following recurrence using intelligent guesswork method. [5]

$$f(n) = \begin{cases} 0 & ,\text{if } n = 0 \\ n^3 + f(n-1) & ,\text{if } n > 0 \end{cases}$$

OR

Que.2 [A] Solve the following recurrence using inhomogeneous recurrence method. [5]

$t_n = 2t_{n-1} + (n+5)\,3^n$

[B] Solve the following recurrence using change of variable method. [6]

$$T(n) = \begin{cases} 1 & ,\text{if } n = 1 \\ 3T\left(\frac{n}{2}\right) + n & ,\text{if } n \text{ is a power of } 2, n > 1 \end{cases}$$

Que.3 [A] Solve the recurrence using master theorem method: $T(n) = 3T(n/4) + n \lg n$ [4]

[B] Solve the recurrence using recursion tree method: $T(n) = 4T(n/2) + n^3$ [4]

[C] Prove that $2^{n+1} = O(2^n)$ but $2^{2n} \neq O(2^n)$ [4]

SECTION – II

Que.4 [A] Explain divide & conquer method using merge sort algorithm and solve recurrence relation of merge sort.

[B] Briefly explain breadth first search with example.

OR

Que.4 [A] Solve recurrence relation of quick sort algorithm for worst case. On what kind of input the worst case of quick sort occurs? How it can be solved using randomized version of quick sort? [6]

[B] Explain use of branch & bound technique for solving assignment problem. [4]

[C] Differentiate between class P and class NP problems. [2]

Que.5 [A] Apply dynamic programming algorithm to construct table for change of amount 11 with coins of denomination 1, 5, 6 and 8. [3]

[B] Write greedy algorithm for fractional knapsack problem. Apply it to solve fractional knapsack problem as shown in Table 1 where p is profit and w is weight of each item and M is knapsack capacity. Take M=30. [7]

Table 1

n	n_1	n_2	n_3	n_4
p_i	50	140	60	60
w_i	5	20	10	12

[C] How to use adjacency list for graph representation? [1]

OR

Que.5 [A] On which kind of problems dynamic programming can be applied? Explain memoization with example. [5]

[B] What is minimum spanning tree? Apply kruskal's algorithm on the graph as shown in Fig.1 to construct minimum spanning tree and write its time complexity. [6]

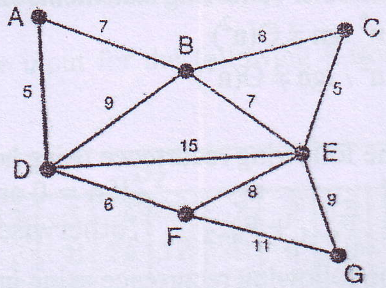

Fig. 1

Que.6 [A] For the following matrices find the order of parenthesization for the optimal chained multiplication using dynamic programming. [7]
Matrices: P: 15 x 5, Q: 5 x 10, R: 10 x 20, S: 20 x 25

[B] Show the working of counting inversion algorithm on given input sequence and find total number of inversions: 11, 3, 1, 2, 4 , 14, 9, 7 [3]

[C] Briefly explain greedy algorithm for activity selection problem. [2]

END OF PAPER